Approximation Algorithms for Minimizing Congestion in Demand-Aware Networks

Wenkai Dai*, Michael Dinitz†, Klaus-Tycho Foerster‡, Long Luo§ and Stefan Schmid¶
*Faculty of Computer Science and UniVie Doctoral School Computer Science DoCS, University of Vienna, Austria
†Department of Computer Science, Johns Hopkins University, USA
‡Department of Computer Science, TU Dortmund, Germany
§University of Electronic Science and Technology of China, P.R. China
¶TU Berlin, Germany and University of Vienna, Austria

***Abstract*—Emerging reconfigurable optical communication technologies allow to enhance datacenter topologies with demand-aware links optimized towards traffic patterns. This paper studies the algorithmic problem of jointly optimizing topology and routing in such demand-aware networks to minimize congestion, along two dimensions: (1) splittable or unsplittable flows, and (2) whether routing is segregated, i.e., whether routes can or cannot combine both demand-aware and demand-oblivious (static) links.**

For splittable and segregated routing, we show that the problem is generally 2-approximable, but APX-hard even for uniform demands induced by a bipartite demand graph. For unsplittable and segregated routing, we establish upper and lower bounds of $O(\log m/\log\log m)$ and $\Omega(\log m/\log\log m)$, respectively, for polynomial-time approximation algorithms, where m is the number of static links. We further reveal that under un-/splittable and non-segregated routing, even for demands of a single source (resp., destination), the problem cannot be approximated better than $\Omega\left(\frac{c_{\max}}{c_{\min}}\right)$ unless P=NP, where $c_{\max}$ (resp., $c_{\min}$) denotes the maximum (resp., minimum) capacity. It remains NP-hard for uniform capacities, but is tractable for a single commodity and uniform capacities.

Our trace-driven simulations show a significant reduction in network congestion compared to existing solutions.

I. Introduction

The popularity of data-centric applications related to e.g., business, entertainment, or artificial intelligence, let to an explosive growth of communication traffic, especially inside datacenters. Over the last years, great efforts have hence been put into the design of novel and more efficient datacenter network designs. A particularly intriguing architecture is based on emerging optical communication technologies, allowing to optimize the network topology towards the traffic demand. Such demand-aware networks are attractive as they allow to leverage the spatial and temporal structure of workloads. More specifically, emerging demand-aware networks, whose topologies are typically hybrid, in that a static (and demand-oblivious) network is enhanced with reconfigurable (and demand-aware) links, introduce unprecedented flexibility in adapting the network topology towards the current traffic demands. In such hybrid networks, the reconfigurable links are usually enabled by optical circuit switches [1]–[3], and particularly, each optical circuit switch provides reconfigurable links by establishing connections between pairs of its ports, i.e., a *matching*.

Extensive past works studied the question of how to jointly optimize topology and routing of such reconfigurable (hybrid) networks [4] for different networking performance metrics, e.g., latency [5], throughput [6]–[11], routing length [12]–[15], flow times [16], [17] etc. Interestingly, *min-congestion*, a most central performance metric in traditional networks, is still not well-understood in reconfigurable networks. Avin et al. [18] and Pacut et al. [19] study optimal *bounded-degree* topology designs, however focusing on a static optimization model in purely demand-aware network, to minimize both the route length and the congestion. Dai et al. [20] consider a hybrid model like in our paper, showing that the problem is already NP-hard for *splittable* (resp., *unsplittable*) and *segregated* (resp., *non-segregated*) routing models when the static network is a tree of height at least two, but tractable for static networks of star topologies. Zheng et al. [21] introduced a greedy-based heuristic algorithm for our *segregated* model but on specific topologies of datacenters. However, not much more is known w.r.t. corresponding approximation bounds, which motivates our study.

In this paper, we are interested in the algorithmic problem underlying such hybrid demand-aware reconfigurable network architectures. In particular, we study the question of how to jointly optimize the topology and the routing in demand-aware networks, with the goal of *minimizing congestion*. We study two different routing models commonly applied in reconfigurable networks, namely whether routing is *segregated* or not, i.e., whether or not flows either have to use exclusively either the static network or the reconfigurable connections. We also consider both splittable and unsplittable flows.

A. Our Contributions

We initiate the study of approximation algorithms for minimizing congestion in hybrid demand-aware networks (for a given matrix of demands). Our results include an overview of approximation results and complexity characterizations in

This project is supported by the European Research Council (ERC), grant agreement No. 864228 (AdjustNet) Horizon 2020, 2020-2025, and supported in part by NSF grants CCF-1909111 and CCF-2228995. Long Luo is supported by the National Natural Science Foundation of China (62102066), National Key Research and Development Program of China (2023YFB2904600), and Young Elite Scientists Sponsorship Program by CAST (2022QNRC001).

TABLE I
SUMMARY OF OUR APPROXIMATION UPPER AND LOWER BOUNDS ON THE MCRN PROBLEM (DEFINITION 1).

Approximation Bounds & Time Complexity	Splittable Flow	Segregated Routing	Restrictions On Demands	Restrictions On Capacities	Results References
2-approximation	yes	yes			Thm. 1
APX-complete	yes	yes	uniform and bipartite demands		Thm. 4
Tractable	yes	yes	single source (resp., dest.)		Thm. 3
$O(\log m/\log\log m)$-approximation	no	yes			Thm. 2
Lower Bound: $\Omega(\log m/\log\log m)$	no	both			Thm. 5
Lower Bound: $\Omega(c_{\max}/c_{\min})$	both	no	single source (resp., dest.)		Thm. 6
NP-hard	both	no	single source (resp., dest.)	uniform	Thm. 7
Tractable	both	no	single commodity	uniform	Thm. 8

general settings (outlined in Table I), and also a fine-grained algorithmic analysis for restricted cases.

a) Segregated Routing: We provide a mixed-integer programming formulation for segregated and un-/splittable flow models whose LP relaxation can be solved efficiently. For splittable flows, we present a 2-approximation algorithm by a novel deterministic rounding approach, and also prove the APX-hardness even if its demands are *uniform* and the *graph induced by demands* is bipartite. However, we also show that the problem becomes tractable for demands with a single source (resp., destination). For unsplittable flows, we show that the min-congestion reconfigurable network problem cannot be approximated better than the min-congestion multi-commodity unsplittable flow problem (MCMF) [22], but any ρ-approximation algorithm based on rounding techniques for the MCMF problem can be utilized to give a 2ρ-approximation for the reconfigurable network problem. This implies an approximability of $\Theta(\log m/\log\log m)$ for segregated and unsplittable routing, where m is the number of static links.

b) Non-Segregated Routing: Under the splittable (resp., unsplittable) flow model, even for demands of a single source (resp., destination), the problem cannot be approximated better than $\Omega(c_{\max}/c_{\min})$ unless P=NP, where $c_{\max}$ (resp., $c_{\min}$) denotes the maximum (resp., minimum) capacity on all links, and it still remains NP-hard for *uniform capacities*, i.e., $c : \vec{E}\cup\vec{\mathcal{E}} \mapsto \{a\}$ for $a \in \mathbb{R}_{>0}$. However, the problem with uniform capacities becomes efficiently solvable for demands of a single commodity under un-/splittable flow.

Our trace-driven simulations show that our algorithms significantly improve on state of the art methods.

B. Organization

We introduce our formal model and preliminaries in §II. Our algorithmic and hardness results for the segregated model are presented in §III, followed by a study of non-segregated routing in §IV. We then investigate the performance of our algorithms with trace-driven simulations in §V. Lastly, we discuss related work in §VI and conclude in §VII.

II. MODEL AND PRELIMINARIES

We first introduce our network model, demands and routing policies, and then formalize the min-congestion demand-aware network design problem in §II-A. We then provide preliminaries for proving hardness or approximation in §II-B.

Network Model. Let $N = (V, E, \mathcal{E}, c)$ be a *reconfigurable (hybrid)* network [23], [24] connecting n nodes $V = \{v_1, \ldots, v_n\}$ (e.g., top-of-the-rack switches), using static links E (usually electrically packet-switched), where (V, E) is called the *static network* of N. The network N also contains a set of reconfigurable (usually optical) links $\mathcal{E}$, s.t., $(V, \mathcal{E})$ constitutes a *complete graph* on V. The graph $(V, E \cup \mathcal{E})$ is a *bidirected*[1] *(multi)-graph* such that two directions of each bidirected link $\{v_i, v_j\} \in E$ (resp. $\{v_i, v_j\} \in \mathcal{E}$), where $v_i, v_j \in V$, work as two *(anti-parallel) directed links* (v_i, v_j) and (v_j, v_i) respectively. We use the symbol $\vec{E}$ (resp. $\vec{\mathcal{E}}$) to denote the set of corresponding directed links of E (resp. $\mathcal{E}$). Moreover, a function $c : \vec{E}\cup\vec{\mathcal{E}} \mapsto \mathbb{R}_{\geq 0}$ defines *capacities* for both directions of each bidirected link in $E \cup \mathcal{E}$, where the *maximum* (resp., *minimum*) *capacity* is denoted by $c_{\max}$ (resp., $c_{\min}$). We denote *uniform capacities* if $c : \vec{E} \cup \vec{\mathcal{E}} \mapsto \{a\}$, where $a \in \mathbb{R}_{\geq 0}$, e.g., $a = 1$. The reconfigurable network N can only implement a subset of reconfigurable links $\mathcal{E}$, which must be a matching $M \subseteq \mathcal{E}$, to provisionally enhance the static network (V, E). The enhanced graph $N(M) = (V, E \cup M, c)$ determines the actual topology of the communicating network, where $N(M)$ is called a *reconfigured network* and the matching M is called a *reconfiguration* of N.

Traffic Demands. The reconfigured network should serve a certain communication pattern, represented as a $|V| \times |V|$ communication matrix $D := (d_{i,j})_{|V|\times|V|}$ (*demands*) with non-negative real-valued entries. An entry $d_{i,j} \in \mathbb{R}_{\geq 0}$ represents the traffic load (frequency) or a demand from the node v_i to the node v_j. With a slight abuse of notation, let $D(v_i, v_j)$ also denote a demand from v_i to v_j hereafter. For each demand $D(v_i, v_j) > 0$, the pair (v_i, v_j) is called a commodity with the source v_i and the destination v_j. The matrix of demands D is called *single commodity* if it contains only one commodity, otherwise multi-commodity. A multi-commodity matrix D is called single source (resp., destination) if all commodities share the same source (resp., destination). Demands D are called *uni-*

[1] Symmetrical connectivity is the standard industry assumption for static cabling, however for reconfigurable links as well. Outside highly experimental hardware, e.g. [5], off-the-shelf products use full-duplex connections [25], [26] and this model assumption is hence prevalent, even in Free-Space Optics [27] proposals.

form if all non-zero entries $d_{i,j} > 0$ have the same value. The graph G_D *induced by* demands D is defined as a simple graph $G_D = (V, E_D)$, where $E_D = \{\{v_i, v_j\} : d_{i,j} + d_{j,i} > 0\}$.

Routing Models. For networking, *unsplittable* routing requires that all flows of a demand $d_{i,j} \in D$ must be sent along a single (directed) path, while *splittable* routing does not restrict the number of paths used for the traffic of each demand; For a reconfigured network, segregated routing requires flows of each demand $d_{i,j} \in D$ being transmitted on either the static network or the reconfigurable link between two endpoints of this demand, but non-segregated routing admits reconfigurable links used as shortcuts for flows along static links [2], [3].

Hence, there are *four different routing models*: *Unsplittable & Segregated (US)*, *Unsplittable & Non-segregated (UN)*, *Splittable & Segregated (SS)*, and *Splittable & Non-segregated (SN)*.

A. Min-Congestion Reconfigurable Network Problem

> "*As minimizing the maximum congestion level in all links is a desirable feature of DCNs* [28], [29]*, the objective of our work is to minimize the maximum link load*"
>
> Yang et al. [30] (ACM SIGMETRICS 2020)

Congestion. Given a reconfigured network $N(M)$ and demands D, let $f : \vec{E} \cup \vec{M} \mapsto \mathbb{R}_{\geq 0}$ be *a flow* serving *all* demands D in $N(M)$, s.t. the size of the net flow from $v_i \in V$ to $v_j \in V$ equals to $d_{i,j}$ for each demand entry $d_{i,j} \in D$, under a routing model $\tau \in \{\text{US}, \text{UN}, \text{SS}, \text{SN}\}$. For each $d_{i,j} \in D$, let $f_{i,j} : \vec{E} \cup \vec{M} \mapsto \mathbb{R}_{\geq 0}$ denote the sub-flow of f caused by the demand d_{ij}. Thus, the flow f can be further defined as $f = \{f_{i,j} : d_{i,j} \in D\}$, where $\forall e \in \vec{E} \cup \vec{M} : f(e) = \sum_{d_{i,j} \in D} f_{i,j}(e)$ and $f(e)$ can exceed $c(e)$. We consider *loads of a flow* f on both static and reconfigurable links, i.e., $\ell : \vec{E} \cup \vec{M} \mapsto \mathbb{R}_{\geq 0}$. *The load of each directed link* $e \in \vec{E} \cup \vec{M}$ induced by a flow f is defined as $\ell(e) := \frac{f(e)}{c(e)}$, and the *maximum load* of f is defined as $\ell_{\max}(f) := \max\left\{\ell(e) : e \in \vec{E} \cup \vec{M}\right\}$. Given a routing model $\tau \in \{\text{US}, \text{UN}, \text{SS}, \text{SN}\}$, the *congestion* in a reconfigured network $N(M)$ to serve D is defined as

$$\lambda := \min\{\ell_{\max}(f) : \text{a flow } f \text{ serving } D \text{ in } N(M) \text{ under } \tau.\}$$

Definition 1 (Min-Congestion Reconfigurable Network Problem (MCRN))**.** *Given a reconfigurable network $N = (V, E, \mathcal{E}, c)$, a routing model $\tau \in \{US, UN, SS, SN\}$, and a demand matrix D, find a matching (reconfiguration) $M \subseteq \mathcal{E}$, s.t., the congestion λ to serve D in the network $N(M)$ under the routing model τ is minimized.*

B. Preliminaries of Approximation Upper and Lower Bounds

A problem has an *approximation upper bound* α if there exists a polynomial-time α-approximation algorithm to solve it, while a problem has an *approximation lower bound* α' if no polynomial-time algorithm can approximate it better than α' unless P=NP. An approximation factor preserving reduction [22] can be constructed to reveal the approximability between two problems.

We next introduce the classic min-congestion multi-commodity flow problem, which will be used later.

Definition 2 (Min-Congestion Multi-Commodity Flow Problem)**.** *We are given a demand matrix D, a static (directed) network $N = (V, E)$ with the capacity function $c : \vec{E} \mapsto \mathbb{R}_{\geq 0}$, and a routing model (splittable/unsplittable). Our goal is to find a flow:*

$$f = \left\{f_{ij} : \vec{E} \mapsto \mathbb{R}_{\geq 0} \mid d_{ij} \in D\right\},$$

serving D under the given (splittable/unsplittable) routing model, s.t., the maximum load $\max_{e \in \vec{E}} \frac{f(e)}{c(e)}$ can be minimized, where $f(e) = \sum_{d_{ij} \in D} f_{ij}(e)$.

III. Segregated Routing Model

In this section, we start to study the MCRN problem under segregated routing. We first introduce LP-based approximation algorithms for both splittable and unsplittable flow models in §III-A, and then show that the MCRN problem of splittable flow is tractable for demands of a single source (resp., destination) in §III-B. Finally, we discuss approximation lower bounds for the problem under both splittable and unsplittable flow models in §III-C.

A. Approximation Algorithms

Our approximation algorithms are based on solving a linear programming relaxation of the MCRN problem under segregated routing and obtaining a feasible solution by deterministic rounding. Thus, we will first present the ILP formulation of the problem under segregated routing, which is same for both splittable and unsplittable flow.

1) ILP Formulation: To simplify the ILP formulation, we denote each node $v_i \in V$ simply by $i \in V$, e.g., a demand $d_{i,j} \in D$ has source $i \in V$ and sink $j \in V$. For the segregated routing model, we note that each reconfigurable link $\{i, j\} \in M$ indicates that its demands $d_{i,j}$ and $d_{j,i}$ must be solely sent on $\{i, j\} \in \mathcal{E}$, and other demands, whose two endpoints are not connected by a reconfigurable link included in M, will only transfer on the static network (V, E). We can write the min-congestion reconfigurable network problem with $\tau \in \{\text{SS}, \text{US}\}$ as the following mixed-integer linear program (MILP).

$$\min \lambda \tag{1}$$

s.t.

$$\sum_{P \in \mathcal{P}_{i,j}} f_P \geq (1 - z_{i,j}) \cdot d_{i,j} \qquad \forall i, j \in V \tag{2}$$

$$\sum_{P \in \mathcal{P} : e \in P} f_P \leq \lambda \cdot c(e) \qquad \forall e \in \vec{E} \tag{3}$$

$$z_{i,j} \cdot d_{i,j} \leq \lambda \cdot c((i,j)) \qquad \forall (i,j) \in \vec{\mathcal{E}} \tag{4}$$

$$0 \leq \sum_{j \in V} z_{i,j} \leq 1 \qquad \forall i \in V \tag{5}$$

$$f_P \geq 0 \qquad \forall P \in \mathcal{P} \tag{6}$$

$$z_{i,j} = z_{j,i} \qquad \forall i, j \in V \tag{7}$$

$$z_{i,j} \in \{0, 1\} \qquad \forall i, j \in V \tag{8}$$

To interpret this MILP formulation, we give the following notes:

- If a bidirected reconfigurable link $\{i,j\} \in \mathcal{E}$ is included in M, then both directions $(i,j) \in \vec{\mathcal{E}}$ and $(j,i) \in \vec{\mathcal{E}}$ are implemented in $\vec{M}$ simultaneously. Let each decision variable $z_{i,j} \in \{0,1\}$ indicate whether to select the directed reconfigurable link $(i,j) \in \vec{\mathcal{E}}$ into $\vec{M}$ for implementation. Thus, the constraint (7) ensures the simultaneous implementation of both directions for each bidirected reconfigurable link $\{i,j\} \in \mathcal{E}$.
- The variable $\lambda \in \mathbb{R}^+$ indicates the maximum load for all directed links in the reconfigured network $\left(V, \vec{E} \cup \vec{M}\right)$. Our goal is to minimize λ by setting these decision variables $z_{i,j}$.
- For every two nodes $i,j \in V$, let $\mathcal{P}_{i,j}$ denote the set of all simple directed paths from i to j in the static network $\left(V, \vec{E}\right)$. For each (directed) static link $e \in \vec{E}$, let $c(e)$ denote its capacity. For each demand $d_{i,j}$, the variable f_P shows the flow size along a directed path $P \in \mathcal{P}_{i,j}$. Finally, let $\mathcal{P}$ be the collection of all directed paths $\mathcal{P}_{i,j}$ for every two $i,j \in V$, i.e., $\mathcal{P} = \bigcup_{i,j \in V} \mathcal{P}_{i,j}$.
- The constraints (2) ensure that each demand $d_{i,j} \in D$ being sent on the static network $(V, \vec{E})$ if $z_{i,j} = 0$, otherwise $d_{i,j}$ being sent on a directed reconfigurable link $(i,j) \in \vec{M}$.
- The constraints of (3) and (4) bound the maximum load on directed links in $\vec{E} \cup \vec{M}$ by λ.
- For the above MILP formulation, we can relax it into an LP formation in an oblivious way, by changing the integrality constraint $\forall i,j \in V : z_{i,j} \in \{0,1\}$ to $\forall i,j \in V : 0 \le z_{i,j} \le 1$, which results in *an integrality gap* ≥ 2.
- To ensure that all indicator variables $z_{i,j}$ lie in the matching polytope after relaxing $z_{i,j} \in \{0,1\}$ to $0 \le z_{i,j} \le 1$, blossom inequalities defined by (9) are usually required. However, blossom inequalities are unnecessary for our LP since our rounding procedures can always guarantee that the generated feasible solution is a matching.

Blossom Inequalities:

$$\sum_{i,j \in U} z_{i,j} \le \frac{|U|-1}{2} \qquad \forall U \subseteq V : |U| \text{ is odd.} \tag{9}$$

a) Solving LP Relaxation.: After relaxing the above MILP, the corresponding LP relaxation can be solved efficiently, although it contains an exponential number of variables $\{f_P : P \in \mathcal{P}\}$ and constraints (6). There are different ways to do it, but we introduce an intuitive way, often employed in approximating the min-congestion multi-commodity flow problem [22], [31]. The original LP formulation can be transferred to an equivalent LP with *a compact formulation*, where the flow for each demand $d_{i,j} \in D$ is presented on each link, i.e., $\left\{ f_e^{i,j} \in \mathbb{R}_{\ge 0} : e \in \vec{E} \cup \vec{\mathcal{E}} \right\}$, instead of along paths in $\mathcal{P}_{i,j}$, and also satisfies flow conservation for all nodes. This compact LP clearly has a polynomial number of variables and constraints, which can be efficiently solved, while a solution of the compact LP can be transferred to a solution of its original LP with the same load on each link.

b) Deterministic Rounding.: Let $Z_{opt} = \{z_{i,j}\}_{i,j \in V}$ be an optimal solution to the LP relaxation of the above MILP. Let λ_{opt} denote the optimal congestion for the fractional optimal solution Z_{opt}. Let λ_{min} denote the minimized congestion of the original ILP, which has $\lambda_{\text{opt}} \le \lambda_{\text{min}}$. Now, we introduce our idea to obtain a feasible integer solution $\hat{Z} = \{\hat{z}_{ij}\}_{i,j \in V}$ based on rounding each fractional variable $z_{i,j}$ in Z_{opt}.

We first consider the following rounding-up function:

$$\hat{z}_{i,j} = \begin{cases} 1 & \text{if } z_{i,j} > 1/2; \\ 0 & \text{if } z_{i,j} \le 1/2. \end{cases}$$

It is easy to see that $\hat{z}_{i,j} = \hat{z}_{j,i}$ for all $i,j \in V$, since $z_{i,j} = z_{j,i}$ in Z_{opt}. For each $i,j \in V$, we round up its flows f_P for all $P \in \mathcal{P}_{i,j}$ by the following way:

$$\hat{f}_P = \begin{cases} f_P / \left(1 - z_{i,j}\right), & \text{if } \hat{z}_{i,j} = 0; \\ 0 & \text{if } \hat{z}_{i,j} = 1. \end{cases}$$

2) Approximation Results: Next, we will show several approximation results based on the above rounding-up method.

Theorem 1. *When $\tau =$ SS, the min-congestion reconfigurable network problem has a polynomial-time 2-approximation algorithm.*

Proof: Due to the degree bound of one, without violating (5), each node $i \in V$ can have at most one indicator variable $z_{i,j} \in Z_{\text{opt}}$ that has $z_{i,j} > 1/2$, where $(i,j) \in \vec{\mathcal{E}}$. Thus, by rounding up the variable $z_{i,j} > 1/2$ to one, each node $i \in V$ can have at most one directed reconfigurable link $(i,j) \in \vec{M}$, which implies that (5) is still satisfied by $\hat{Z}$. For each $i,j \in V$, if $\hat{z}_{i,j} = 1$ then the constraint (2) is clearly satisfied as $0 = 0$; otherwise, by summing flows $\hat{f}_P$ for all $P \in \mathcal{P}_{ij}$, we can have the inequality (10), still satisfying (2).

$$\begin{aligned} \sum_{P \in \mathcal{P}_{i,j}} \hat{f}_P = \sum_{P \in \mathcal{P}_{i,j}} \frac{f_P}{(1 - z_{i,j})} &= \frac{1}{(1 - z_{i,j})} \sum_{P \in \mathcal{P}_{i,j}} f_P \\ &\ge \frac{(1 - z_{i,j})\, d_{i,j}}{(1 - z_{i,j})} = d_{i,j} \end{aligned} \tag{10}$$

For each directed static link $e \in \vec{E}$, we consider the constraint (3). Note that $\hat{f}_P > 0$ holds, where $P \in \mathcal{P}_{i,j}$ and $i,j \in V$, only if $z_{i,j} \le 1/2$, which implies $\hat{f}_P = f_P/(1 - z_{i,j}) \le 2 f_P$. Thus, we have the following inequation.

$$\sum_{P \in \mathcal{P} : e \in P} \hat{f}_P \le \sum_{P \in \mathcal{P} : e \in P} 2 \cdot f_P \le 2 \cdot \lambda_{\text{opt}} \cdot c(e) \tag{11}$$

Regarding the constraint (4), for each $(i,j) \in \vec{\mathcal{E}}$, it implies

$$\hat{z}_{i,j} \cdot d_{i,j} \le 2 \cdot z_{i,j} \cdot d_{i,j} \le 2 \cdot \lambda_{\text{opt}} \cdot c\left((i,j)\right).$$

Now, it is safe to conclude that $\hat{Z}$ is a feasible integer solution, and $f := \left\{ \hat{f}_P : P \in \mathcal{P}_{i,j} \text{ and } i,j \in V \right\}$ defines a feasible

flow serving all demands D. As $\lambda_{\text{opt}} \leq \lambda_{\min}$, it implies that our deterministic rounding-up method can achieve a 2-approximation ratio within $O\left(n^2\right)$. The exact runtime of the algorithm depends on the specific LP formulation and LP solver, but is polynomial. ■

We will now extend the above method to obtain an approximation result for the MCRN problem of $\tau = \text{US}$.

Theorem 2. *If the min-congestion multi-commodity unsplittable flow problem has a ρ-approximation algorithm based on rounding techniques on its LP solution, e.g., $\rho = O(\log m / \log\log m)$ [32], then the MCRN problem with $\tau = US$ can be approximated by 2ρ.*

Proof: First, we note that the ILP formulation in §III-A1 also works for $\tau = \text{US}$. Let λ_{opt} be the optimal value of the above relaxed LP for $\tau = \text{SS}$, which is obviously a lower bound for the MRCP problem with $\tau = \text{US}$.

For the MCRN problem of $\tau = \text{US}$, we first solve it by assuming $\tau = \text{SS}$ as Theorem 1 to obtain a matching M. For each $\{i, j\} \in M$, we set $d_{i,j} = 0$ and $d_{j,i} = 0$ in D to obtain a new set of demands D'. By replacing D by D' into the ILP formulation in §III-A1, we can obtain a new relaxed LP formulation $LP(D')$, which has an optimal congestion λ. Clearly, it has $\lambda \leq 2 \cdot \lambda_{\text{opt}}$ by Theorem 1. Given D', we can solve the min-congestion multi-commodity unsplittable flow problem on a static network (V, E) to obtain a ρ-approximation congestion λ' by applying rounding techniques on the splittable (optimal) flow of $LP(D')$, which further implies $\lambda' \leq \rho \cdot \lambda \leq 2\rho \cdot \lambda_{\text{opt}}$. Thus, λ' is a 2ρ-approximation result for $\tau = \text{US}$. ■

B. Polynomial-Time Solvable Cases

After obtaining approximation results for general cases of segregated model, we will show a restricted case, where demands contain a single source (resp., dest.), is efficiently solvable in Theorem 3. The proof of Theorem 3 is easy as only one reconfigurable link can be used here under segregated model, which is deferred due to limited space.

Theorem 3. *When $\tau = SS$, if the demands D have a single source (resp., dest.), the MCRN problem is polynomial-time solvable.*

C. Approximation Lower Bounds for Segregated Routing Model

In the following, we show that the problem under the splittable and segregated routing is APX-complete, implying that our 2-approximation algorithm achieves a tight bound.

Theorem 4. *For $\tau = SS$, the min-congestion reconfigurable network problem (MCRN) is APX-complete, even if the demands D are uniform in size and the graph G_D induced by D is acyclic and bipartite, e.g., a 3D matching.*

Proof: Let Π^{con} denote the min-congestion reconfigurable network problem (MCRN) of $\tau = \text{SS}$, where the induced graph G_D by demands D is acyclic and bipartite, and $\forall d_{i,j} \in D : d_{i,j} \in \{3, 0\}$[2]. We note that the minimum vertex cover problem on 3-regular graph Π is APX-complete, which cannot be approximated better than a ratio $\rho \in (1, 2)$ [33]. To show APX-hardness of Π^{con}, we give a gap-preserving reduction from the minimum vertex cover problem on 3-regular graph to reveal that Π^{con} cannot be approximated better than a factor $\min\{\rho, 1.2\}$. The construction is illustrated in Fig. 1.

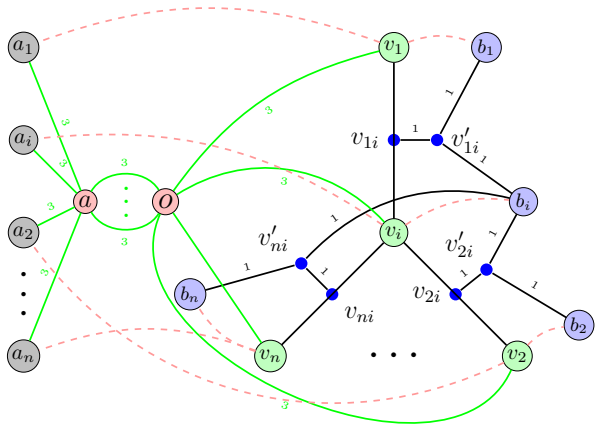

Fig. 1. Illustration of our gap-preserving reduction from the minimum vertex cover problem to the MCRN problem with $\tau = \text{SS}$. Each green node $v_1, v_2, v_i, \ldots, v_n$ corresponds to one vertex in the given minimum vertex cover instance. The dashed lines show a subset of reconfigurable links $\mathcal{E}$, whose two endpoints have non-zero demands in D, and solid lines indicate static links E, where there are k parallel paths between a and o in the static network. Each bi-directed link in $\vec{\mathcal{E}} \cup E$ has the same capacity in both directions, and we define capacities: $c : \vec{\mathcal{E}} \mapsto \{3\}$ and $c : \vec{E} \mapsto \{1, 3\}$. The capacity value is marked on each bi-directed (static) link in the figure.

Given an instance $I = (G_U = (U, E_U))$ of the minimum vertex cover problem Π, where G_U is a 3-regular graph with $|U| = n$, we construct an instance I' of Π^{con}, as shown in Figure 1. Let the constructed instance be $I' = (G, \mathcal{E}, c, D)$, where a static network $G = (V, E)$, a set of reconfigurable links $\mathcal{E}$, the capacity function $c : \vec{E} \cup \vec{\mathcal{E}} \mapsto \{1, 3\}$ (each bi-directed link has the same capacities on both directions), and demands D. We construct nodes V as follows:

- For each vertex $u_i \in U$ of I, where $i \in \{1, \ldots, n\}$, we construct three nodes: $v_i \in V_1$, $a_i \in A$ and $b_i \in B$;
- For each edge $\{u_i, u_j\} \in E_U$, we construct 2 nodes: $v_{ij} \in V_2$ and $v'_{ij} \in V_2$;
- Moreover, we have additional two nodes $\{a, o\}$.
- Finally, let constructed nodes be

$$V = A \cup B \cup V_1 \cup V_2 \cup \{a, o\}.$$

All static links (edges) E will be constructed as follows, where their capacities are defined by $c : \vec{E} \mapsto \{1, 3\}$:

- For each edge $\{u_i, u_j\} \in E_U$ of I, where $i, j \in \{1, \ldots, n\}$ and $i \neq j$, we construct the following static (bi-directed) links: $\{v_i, v_{ij}\} \in E_1$, $\{v_{ij}, v_j\} \in E_1$, $\{v_{ji}, v'_{ij}\} \in E_2$, $\{v'_{ij}, b_j\} \in E_3$ and $\{v'_{ij} b_i\} \in E_3$
- For each $u_i \in U$ of I, there are two static bi-directed links: $\{a_i, a\} \in E_A$ and $\{v_i, o\} \in E_o$.

[2] To have integer capacities, in this proof, we use a matrix of entries in $\{0, 3\}$, which can be transferred to a $0/1$ matrix by scaling down each capacity by a factor 3.

- There are k paralleling links between a and o, i.e., $E_{a,o} = \{\{a,o\}_i : 1 \leq i \leq k\}$.
- Finally, let $E = E_1 \cup E_2 \cup E_3 \cup E_A \cup E_o \cup E_{a,o}$, and $\forall \{u,v\} \in E_1 \cup E_2 \cup E_3 : c((u,v)) = c((v,u)) = 1$, otherwise $c((u,v)) = c((v,u)) = 3$.

Moreover, we define demands D as follows: for each $i \in \{1,\ldots,n\}$, it has demands $D(v_i,b_i) = \delta_{v_i}$ and $D(v_i,a_i) = 3$, where $\delta_{v_i} = 3$ denotes the degree of u_i in G_U of I. We note that the graph G_D induced by demands D in our construction is also a 3D-matching.

Finally, for every $u,v \in V$, there is a reconfigurable (bi-directed) link $\{u,v\}$ with capacities: $c((u,v)) = c((v,u)) = 3$. However, due to the splittable and segregated model ($\tau =$ SS), only reconfigurable links $\{u,v\}$ that have $D(u,v)+D(v,u) > 0$ need to be considered for reconfiguration.

Next, we will prove:

- If G_U has a vertex cover $U^* \subseteq U$ with $|U^*| \leq k$, then the constructed instance I' has the minimized congestion $\lambda \leq 1$;
- If any vertex cover $U^* \subseteq U$ in G_U has $|U^*| \geq \rho \cdot k$, where the ratio $\rho \in (1,2)$, then the constructed instance I' has the minimized congestion $\lambda \geq \min\{\rho, 1.2\}$.

We first do some analysis of the constructed instance I'. Due to $b = 1$ and $|V_1| = n$, the reconfiguration (matching) M can contain at most n reconfigurable links. Let $M_1 \subseteq M$ denote the selected reconfigurable links between V_1 and A and $M_2 \subseteq M$ denote the selected reconfigurable links between V_1 and B, where $M_1 \cap M_2 = \emptyset$ and $M_1 \cup M_2 = M$.

We first claim that the set of nodes $V^* \subseteq V_1$ contained in M_2 must imply a vertex cover U^* in the graph G_U, where $\forall v_i \in V^* \implies u_i \in U^*$, otherwise, the maximum load factor is $\lambda \geq 1.2$.

If V^* implies a vertex cover $U^* \subseteq U$ in G_U, it is easy to note that the maximum load on the static edges in $E_1 \cup E_2 \cup E_3$ is at most one. If $v_i \in V^*$, then its demand $D(v_i,b_i) = 3$ is sent on the reconfigurable (directed) link (v_i,b_i) with $c((v_i,b_i)) = 3$, otherwise $D(v_i,b_i) = 3$ must go through the corresponding three paralleling paths consisting of static links: $(v_i, v_{ij}, v'_{ij}, b_i)$, where each j indicates an edge $\{u_i,u_j\} \in E_U$ in G_U.

Reversely, we assume that V^* does not imply a vertex cover in the graph G_U. We know at least one edge $\{u_i,u_j\}$ is not covered in G_U, which also means $v_i, v_j \notin V^*$. There are at most 5 edge-disjoint paths between $\{v_i,v_j\}$ and $\{b_i,b_j\}$ on static network G, which provides total capacities of 5. Since $D(v_i,b_i) = 3$ and $D(v_j,b_j) = 3$, then 6 units of demands need to be sent on these 5 edge-disjoint paths, which indicates the maximum load $\lambda \geq 1.2$.

After showing that M_2 must imply a vertex cover U^* in G_U, otherwise $\lambda \geq 1.2$, then we discuss the relationship between the size $|M_2|$ and the maximum load λ. If G_U has a minimum vertex cover U^* of the size k, then the load on any static link in $E_o \cup E_a \cup E_{a,o}$ is at most one, which implies the congestion $\lambda \leq 1$. But if all vertex covers having size $\geq \rho \cdot k$ in G_U, where $\rho \in (1,2)$, then it must have the congestion $\lambda \geq \rho$.

Given an arbitrary vertex cover $U^* \subseteq U$ in G_U, where $|U^*| \geq \rho \cdot k$, let $V^* \subseteq V$ denote the corresponding nodes in G with $|V^*| \geq \rho \cdot k$. Since each $v_i \in V^*$ must be included in the reconfigurable links M_2 not M_1, then its demand $D(v_i,a_i) = 3$ must be sent through static links. Thus, there are at least $3 \cdot \rho \cdot k$ units of demands that must be transferred from o to a. Furthermore, there are k paralleling static links $E_{a,o}$ between a and o, where $c : \vec{E}_{a,o} \mapsto 3$. Clearly, the maximum load λ on directed links in $E_{a,o}$ is at least ρ.

Since the minimum vertex cover problem on cubic graphs is APX-complete [33], there must exist a $\rho \in (1,2)$ s.t., the minimum vertex cover problem on cubic graphs cannot be approximated better than ρ unless $P = NP$. Therefore, the above reduction implies that our problem cannot be approximated better than $\min\{1.2,\rho\}$, which further implies APX-hardness. Since our problem can be approximated by two, then it is in APX-complete. ∎

It remains to investigate the approximation lower bound for unsplittable and segregated routing. In Theorem 2, we reduce the MCRN problem with $\tau =$ US into a sub-problem, which belongs to the min-congestion multi-commodity unsplittable flow problem. Now, we can formally show the connection between these two problems in terms of approximability.

Theorem 5. *There is an approximation factor-preserving reduction from the min-congestion of multi-commodity unsplittable flow problem to the MCRN problem with $\tau =$ US. Thus, the lower bound $\Omega(\log m / \log\log m)$ [34] on approximating the min-congestion of multi-commodity unsplittable flow problem is also an approximation lower bound for the MCRN problem with $\tau =$ US.*

Proof: We prove it by giving a factor-preserving reduction. Given an instance $I = (V,E,c,D)$ of the min-congestion of multi-commodity unsplittable flow problem (Definition 2), we will construct an instance $I' = (V',E',\mathcal{E}',c',D')$ of the MCRN problem with $\tau =$ US. W.L.O.G., we assume that the minimum value in the capacity function c is one.

We first construct a copy of I contained in I', s.t., $V \subset V'$, $E \subset E'$, $c \subset c'$ and $D \subset D'$. In addition, for each node $v_i \in V$, we create a node $v_{i'} \in V'$ and a demand $d_{i,i'} \in D'$ with $d_{i,i'} = \alpha$, where $\alpha > 0$ is a very large number. For each node $v_i \in V$, we construct a reconfigurable edge $\{i,i'\} \in \mathcal{E}'$ with capacities $c'((i,i')) = \alpha$ and $c'((i',i)) = 1$, and a static link $\{i,i'\} \in E'$ having the capacity one on both directions. We finish the construction of I' after setting the capacities of other unmentioned reconfigurable links in $\mathcal{E}'$ by one.

Since α is very large in the instance I', for each node $v_i \in V$, its reconfigurable edge $\{i,i'\} \in \mathcal{E}$ must be included in the matching M, otherwise the min-congestion in I' after reconfiguration can be as large as α. After that, we cannot add any reconfigurable link into the matching M of I'. Now, the instance I' equals to the given instance I of the multi-commodity flow problem. Therefore, I' cannot be approximated better than I. ∎

IV. Non-Segregated Routing Model

In this section, we study the MCRN problem under non-segregated routing, where we are particularly interested in un-

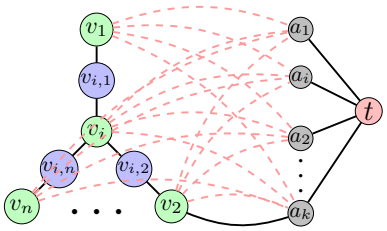

Fig. 2. Illustration of our gap-producing reduction from the min-vertex cover problem to the MCRN problem in the non-segregated model. The figure shows an instance of the MCRN, where each green node $v_1, v_2, v_i, \ldots, v_n$ corresponds to one vertex in the min-vertex cover instance.

derstanding restricted cases, such as, demands of single source (resp., destination), and single commodity, uniform capacities, and their combinations. We will first show approximation lower bounds and NP-hardness for uniform capacities in §IV-A, and then introduce some tractable restricted cases in §IV-B.

A. Approximation Lower Bounds for Non-Segregated Routing

By Theorem 6, we show that the MCRN problem under non-segreated routing has an approximation lower bound $\Omega\left(\frac{c_{\max}}{c_{\min}}\right)$ for both splittable and unsplittable flow, which is pessimistic since the value of $\frac{c_{\max}}{c_{\min}}$ can be arbitrarily large. We remark that the lower bounds $\Omega\left(\frac{c_{\max}}{c_{\min}}\right)$ still holds if each link has the same capacity on both directions, which can be derived by adapting the given proof.

Theorem 6. *When $\tau \in \{SN, UN\}$, given demands D of a single source (resp., destination), the MCRN problem cannot be approximated better than $\Omega\left(\frac{c_{\max}}{c_{\min}}\right)$ unless $P = NP$.*

Proof: We give a proof for demands of a single destination, and the case of a single source can be shown symmetrically. We give a gap-producing reduction from a decision problem of vertex cover Π on 3-regular graphs to the MCRN problem Π'. For an instance $I = (G_U = (U, E_U), k)$ of Π, where $|U| = n$, we will construct an instance I' of Π', illustrated in Fig. 2.

Let the constructed instance be $I' = (G, \mathcal{E}, c, D)$, where $G = (V, E)$ is the static network, a set of reconfigurable links $\mathcal{E}$, a capacity function $c : \vec{E} \cup \vec{\mathcal{E}} \mapsto \{1, \epsilon\}$, s.t., $c_{\max} = 1$ and $c_{\min} = \epsilon$ ($0 < \epsilon \ll 1$), and a demand matrix D.

The set of nodes V in I' are constructed as follows:

- For each vertex $u_i \in U$ of I, where $i \in \{1, \ldots, n\}$, we construct a node $v_i \in V_1$;
- For each $i \in \{1, \ldots, k\}$, we have a node $a_i \in A$;
- For each edge $\{u_i, u_j\} \in E_U$, we construct a node: $v_{i,j} \in V_2$;
- Moreover, we have additional one node t, and let

$$V = A \cup V_1 \cup V_2 \cup \{t\}.$$

Static (bidirected) links E and their capacities $c : \vec{E} \mapsto \{1, \epsilon\}$ are constructed as follows:

- For each edge $\{u_i, u_j\} \in E_U$ of I, where $i, j \in \{1, \ldots, n\}$ and $i \neq j$, we construct two static (bidirected) links: $\{v_i, v_{i,j}\} \in E_1$ and $\{v_{i,j}, v_j\} \in E_1$ with the capacities defined as:

$$c\left((v_i, v_{i,j})\right) = \epsilon, \qquad c\left((v_j, v_{i,j})\right) = \epsilon,$$
$$c\left((v_{i,j}, v_i)\right) = 1, \quad \text{and } c\left((v_{i,j}, v_j)\right) = 1.$$

- For each $i \in \{1, \ldots, k\}$, we have a static (bidirected) link $\{a_i, t\} \in E_t$, s.t., $c\left((a_i, t)\right) = 1$ and $c\left((t, a_i)\right) = \epsilon$.
- Since the static network (V, E) is a connected graph, we construct a static link e^* connecting a node in V_1 to another node in A, e.g., $e^* = \{v_2, a_k\}$ in Fig. 2.
- Finally, let $E = E_1 \cup E_t \cup \{e^*\}$, where every static link has a capacity ϵ on both directions unless otherwise specified.

For any two nodes in V, there is a reconfigurable (bidirected) link in $\mathcal{E}$. For each $i \in \{1, \ldots, n\}$ and $j \in \{i, \ldots, k\}$, there is one reconfigurable links $\{v_i, a_j\} \in \mathcal{E}$ with capacities $c\left((v_i, a_j)\right) = 1$ and $c\left((a_j, v_i)\right) = \epsilon$. For other reconfigurable links, they have a capacity ϵ on both directions.

We complete the construction by giving demands D as follows: for each edge $\{u_i, u_j\} \in E_U$ of I, we have a node $v_{i,j} \in V$ with a demand $D(v_{i,j}, t) = 1/3$.

Next, we show that, when $\tau \in \{\text{SN}, \text{UN}\}$, if I has a vertex cover of size $\leq k$, then the minimized congestion λ of I' is at most 1; and if I has a vertex cover of size $> k$, then the minimized congestion λ of I' has $\lambda \geq \frac{1}{12\epsilon}$ for $\tau = \text{SN}$ (resp., $\lambda \geq \frac{1}{3\epsilon}$ for $\tau = \text{UN}$).

Clearly, we need to find k nodes in V_1 to connect k nodes in A by reconfigurable links (matching) M, s.t., each demand can be sent to t without going through any directed link of the capacity ϵ.

First, if G_U has a vertex cover $U' \subseteq U$ of size k, then for each $u_i \in U'$, we add the corresponding reconfigurable link $\{v_i, a_i\} \in \mathcal{E}$ into M. By this reconfiguration M, each node $v_{i,j}$ can find a directed path to t, s.t., each directed link in this path has the capacity one, then $\lambda \leq 1$.

On the other hand, if G_U does not have a vertex cover of size $\leq k$, then for any k vertices $U' \subseteq U$, there exists an edge $\{u_i, u_j\} \in E_U$, s.t., $u_i \notin U'$ and $u_j \notin U'$. Thus, the demand $D(v_{i,j}, t)$ cannot be sent on reconfigurable links incident on either v_i or v_j. It has to find another node $v_f \in V_1$, which has a reconfigurable link $\{v_f, a_l\}$ included in a directed path (v_f, a_l, t), where $a_l \in A$. W.L.O.G., we assume $\{u_f, u_i\} \in E_U$. Then, any directed path from $v_{i,j} \in V_1$ to $v_f \in V_1$ must visit the directed link $(v_i, v_{i,f})$ that has the capacity ϵ. Thus, when $\tau = \text{SN}$ (resp., $\tau = \text{UN}$), it implies $\lambda \geq \frac{1}{12\epsilon}$ (resp., $\lambda \geq \frac{1}{3\epsilon}$), where $c_{\max} = 1$ and $c_{\max} = \epsilon$.

Therefore, the I' cannot be approximated within a ratio $\frac{1}{12\epsilon}$, i.e., $\Omega\left(\frac{c_{\max}}{c_{\min}}\right)$, unless $P = NP$. ∎

After knowing the approximation lower bound $\Omega\left(\frac{c_{\max}}{c_{\min}}\right)$ in Theorem 6, an immediate question is if the problem is tractable when $c_{\max} = c_{\min}$. Next, we will negate this conjecture by showing the NP-hardness for splittable (resp., unsplittable) and segregated routing even if it has a single source (resp., dest.) and uniform capacities.

Theorem 7. *When $\tau \in \{SN, UN\}$, the MRCN problem remains NP-hard for demands of a single source (resp., destination) and uniform capacities.*

Proof: We first give a proof for $\tau =$ SN by a many-one reduction from a decision problem Π of the makespan scheduling on identical machines, which is NP-hard [35].

For the makespan scheduling on identical machines [22], we are given a set $\mathcal{M}$ of machines and a set $\mathcal{J}$ of jobs, where each job j has a processing time $p_j \in \mathbb{Z}^+$ on any machine i, and our goal is to schedule the jobs on the machines to minimize the *makespan*, i.e., the maximum completion time.

Given an instance $I = \left(\mathcal{J}, \mathcal{M}, \bigcup_{j\in\mathcal{J}} p_j, \beta\right)$ of Π, which decides if I has a makespan bounded by β, we construct an instance $I' = (V, E, \mathcal{E}, c, D)$ (single destination t) of the MCRN problem Π'. Moreover, let $|\mathcal{J}| = n$ and $|\mathcal{M}| = m$.

For each job $j \in \mathcal{J}$, we construct a source node $v_j \in V$, which has a static link $\{v_j, t\} \in E$ to the single destination $t \in V$ and a demand $D(v_j, t) = \beta + p_j$. For each machine $i \in \mathcal{M}$, we construct a node $u_i \in V$ that has n neighbors $U_i = \left\{u_i^1, \ldots, u_i^n\right\} \subseteq V$ included in the static network, and a static link $\{u_i, t\} \in E$. Moreover, we construct a node $t' \in V$, which has a static link $\{t, t'\} \in E$ and a demand $D(t', t) = 2\beta$. Recall that, for our network model, there exists a reconfigurable link in $\mathcal{E}$ for any two nodes in V but only a subset of $\mathcal{E}$ can be selected into the matching $M \subset \mathcal{E}$. For the uniform capacities, we set $c : \vec{E} \cup \vec{\mathcal{E}} \mapsto \{1\}$.

Next, we need to prove that the optimal congestion λ of I' satisfies $\lambda \leq \beta$ iff the minimum makespan of I is not more than β. The proof details are deferred to the technical report due to space constraint.

For $\tau =$ UN, we can change the construction by replacing each static link $\{v_j, t\} \in E$ with two static links $\left\{v_j, v_j'\right\} \in E$ and $\left\{v_j', t\right\} \in E$ and defining new demands $D(v_j, t) = p_j$ and $D\left(v_j', t\right) = \beta$. Then, the above argument works for the new construction similarly. ∎

B. Polynomial-Time Solvable Cases

In contrast to Theorem 7, Theorem 8 reveals that the MCRN problem with $\tau \in \{\text{SN}, \text{UN}\}$ can become tractable if demands are further restricted to be single-commodity (the proof is deferred due to space constraint).

Theorem 8. *For $\tau \in \{SN, UN\}$, given a single-commodity demand and uniform capacities, the MCRN problem is polynomial-time solvable.*

V. Evaluations

We complement our theoretical analysis with an empirical evaluation of the performance of our algorithms under realistic workloads. We first describe our methodology in §V-A and then discuss our results in §V-B. We will share our implemnetation with the research community together with this paper.

A. Methodology

We employ the following baselines and implemented the corresponding algorithms to compare with our algorithm (labelled as *MC*) under *segregated* and *un/-splittable* models.

Baselines. We first consider a *Maximum Weight Matching* algorithm as a baseline, which aims to maximize the sum of flow quantities on reconfigurable links, also employed by e.g., [2], [3]. Second, we also compare to a *Greedy* approach (labelled as *Greedy*), where we greedily seek a compatible reconfigurable link to reroute the maximum demands in each iteration until the matching cannot be extended further. Similar greedy algorithms are also used by e.g., Halperin et al. [28] and Zheng et al. [21]. Lastly, we additionally plot the congestion on the static network without any reconfigurable link (label: *Oblivious*) and the optimum of the LP before rounding (label: *LP*) as upper and lower bounds respectively.

Traffic Workloads. Since traffic traces in different networks and running different applications can differ significantly [5], [36]–[39], we collected a number of real-world and synthetic traces to generate traffic matrices to evaluate our algorithms. More specifically:

- **HPC traces:** We first employ four traces of exascale applications in High Performance Computing (HPC) clusters [39], [40]: MOCFE, NeckBone, CNS, and MultiGrid.
- **Synthetic traces:** We further consider the synthetic pFabric traces, which are frequently used as benchmarks in scientific evaluations [41]. In a nutshell, we generate demand matrices from workloads that arrive according to a Poisson process between random sources and destinations.

Experimental Setup. We implement the topologies of static networks by generating random k-regular graphs for $k = 4, 8$ (k denotes the number of ports connecting other ToR switches) as Jellyfish [42], and using uniform capacities on both static and reconfigurable links. For unsplittable[3] (resp., splittable) flows, we consider one shortest path (resp., three shortest paths) for each demand for HPC traces (Fig. 3) (resp., pFabric traces (Fig. 4)). We repeat each setting by running it 5 times to obtain averaged results, normalizing the loads on links.

B. Results and Discussion

We summarize our evaluation results in terms of the minimized congestion for these four HPC traces in Fig. 3 and for pFabric traces in Fig. 4.

Clearly, all considered algorithms significantly improve the congestion over the Oblivious baseline, and our algorithms (MC) typically outperform the others. We observe that our MC algorithm can provide the stable benefits throughout all investigated scenarios, e.g., the changing number of nodes, diverse traces and varying average degrees of static networks, while Greedy is worst in achieving a consistent performance.

More specifically, for pFabric traces, our MC algorithm (splittable) can achieve $\approx 65\% - 75\%$ of the original congestion of Oblivious baseline for $40 - 180$ nodes, where its performance is best on 150 nodes and notably decreases when nodes are increased from 140 to 180. The MC algorithm can at least provide a 1.3-approximation w.r.t. the optimal value of our LP.

[3] Due to the large size of networks and demands, in practice, it is usual to restrict the maximum number of paths when computing and implementing splittable flow, as done in, e.g., Jellyfish [42].

(b) $k = 4$ (c) $k = 8$

Fig. 3. Algorithmic comparison of the min-congestion for four traces of exascale applications in HPC clusters, using random k-regular graphs for the static network.

Fig. 4. Algorithmic comparison of the min-congestion for different synthetic traces of pFabric clusters, with random 4-regular graphs as static networks.

Regarding the HPC traces, the MC algorithm (unsplittable) still dominates Greedy and Max. Weight Matching, which keeps a comparatively lower congestion, reducing the congestion of Oblivious by $\approx 30\% - 35\%$, but the variance is slightly higher than in the pFabric traces, matching empirical observations on the complexity of the traces produced by these synthetic traces [39]. Meanwhile, we can observe very similar plots for all algorithms in static networks of $k = 4$ and $k = 8$.

VI. Further Related Work

Reconfigurable networks have recently received much attention both in the theory and the applied community. Reconfigurable networks generally come in two flavors: demand-oblivious networks such as RotorNet [43], Opera [44], Sirius [45], and Mars [10] provide an unprecedented throughput by avoiding (or at least minimizing) multi-hop forwarding compared to their static counterparts [46], [47]. Demand-aware networks such as ProjecToR [5], SplayNets [48], Jupiter [49], Cerberus [8], Eclipse [24], Duo [9], among many other [2]–[4], [16], [17], [43], [50]–[52], are tailored towards skewed and structured workloads and leverage spatial and temporal locality to improve throughput further [38], [39], [53].

However, many of these works revolve around other objectives or network settings, e.g., [5]–[7], [12], [15], [17], [52], [54]–[59], and the important aspect of congestion is still not well understood. We refer the reader to the survey by [4] for a more general overview.

The works by Avin et al. [18] and Pacut et al. [19] study bounded-degree topology designs to minimize route length and congestion, and provide approximation lower bounds and a 6-approximation algorithm for min-congestion for sparse demands. However, their models consider optimizing the topology solely consisting of reconfigurable links without a static network, which fundamentally differs from our model of hybrid networks.

Zheng et al. [21] consider a problem setting similar to our segregated model and study how to enhance classic datacenter network topologies, such as Diamond, BCube and VL2, with small reconfigurable switches. They present NP-hardness results on general graphs, although these results do not transfer to specific data center topologies or trees, respectively, while they also introduce a greedy-based heuristic algorithm without performance guarantees. Yang et al. [30] study how to enhance the datacenter network by 60GHz wireless reconfigurable links for min-congestion under unsplittable and non-segregated routing. However, their congestion definition is structurally different from our model, as they consider undirected links, and furthermore, their reconfigurable links are formed under wireless interference constraints.

Related in name and spirit, is the so-called HYBRID model, introduced by Augustine et al. [60]. So far, the investigations in the HYBRID model focused on, e.g., path length, diameter, and (competitive) routing [60]–[64], and it would be interesting to develop a unifying framework of hybrid demand-aware networks and the HYBRID model.

Dai et al. [20] studied the same network model as we do in this paper, showing that the reconfigurable network-design for the objective of min-congestion is already NP-hard for splittable (resp., unsplittable) and segregated (resp., non-segregated) routing models when the static network is a tree of height at least two, but tractable for static networks of star topologies. However, they only provide algorithms with guarantees for very specific problem instances (i.e., star graphs) and also no approximation hardness lower bounds.

VII. Future Work

Our work leaves open several interesting questions for future research. In particular, it remains to provide a complete picture of tight upper and lower bounds on approximating the non-segregated routing problems.

References

[1] S. Aleksic, "The future of optical interconnects for data centers: A review of technology trends," in *2017 14th International Conference on Telecommunications (ConTEL)*, June 2017, pp. 41–46.

[2] G. Wang *et al.*, "c-through: part-time optics in data centers," in *SIGCOMM*. ACM, 2010, pp. 327–338.

[3] N. Farrington *et al.*, "Helios: a hybrid electrical/optical switch architecture for modular data centers," in *SIGCOMM*. ACM, 2010, pp. 339–350.

[4] M. Nance Hall *et al.*, "A survey of reconfigurable optical networks," *Opt. Switch. Netw.*, vol. 41, p. 100621, 2021.

[5] M. Ghobadi *et al.*, "Projector: Agile reconfigurable data center interconnect," in *SIGCOMM*. ACM, 2016, pp. 216–229.

[6] A. Singla, P. B. Godfrey, and A. Kolla, "High throughput data center topology design," in *NSDI*. USENIX, 2014, pp. 29–41.

[7] N. Devanur *et al.*, "Stable matching algorithm for an agile reconfigurable data center interconnect (MSR-TR-2016-1140)," Microsoft Research, Tech. Rep., June 2016.

[8] C. Griner *et al.*, "Cerberus: The power of choices in datacenter topology design (a throughput perspective)," in *Proc. ACM SIGMETRICS*, 2022.

[9] J. Zerwas *et al.*, "Duo: A high-throughput reconfigurable datacenter network using local routing and control," in *ACM SIGMETRICS*, 2023.

[10] V. Addanki, C. Avin, and S. Schmid, "Mars: Near-optimal throughput with shallow buffers in reconfigurable datacenter networks," in *ACM SIGMETRICS*, 2023.

[11] K. Hanauer *et al.*, "Dynamic demand-aware link scheduling for reconfigurable datacenters," in *IEEE INFOCOM*, 2023.

[12] T. Fenz, K. Foerster, S. Schmid, and A. Villedieu, "Efficient non-segregated routing for reconfigurable demand-aware networks," *Comput. Commun.*, vol. 164, pp. 138–147, 2020.

[13] K. Foerster *et al.*, "Characterizing the algorithmic complexity of reconfigurable data center architectures," in *ANCS*, 2018.

[14] K. Foerster *et al.*, "On the complexity of non-segregated routing in reconfigurable data center architectures," *Comp. Comm. Rev.*, vol. 49(2), pp. 2–8, 2019.

[15] K. Foerster and S. Schmid, "Survey of reconfigurable data center networks: Enablers, algorithms, complexity," *SIGACT News*, vol. 50, no. 2, pp. 62–79, 2019.

[16] M. Dinitz and B. Moseley, "Scheduling for weighted flow and completion times in reconfigurable networks," in *INFOCOM*. IEEE, 2020.

[17] J. Kulkarni, S. Schmid, and P. Schmidt, "Scheduling opportunistic links in two-tiered reconfigurable datacenters," in *33rd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, 2021.

[18] C. Avin, K. Mondal, and S. Schmid, "Demand-aware network design with minimal congestion and route lengths," in *INFOCOM*. IEEE, 2019, pp. 1351–1359.

[19] M. Pacut, W. Dai, A. Labbe, K. Foerster, and S. Schmid, "Improved scalability of demand-aware datacenter topologies with minimal route lengths and congestion," *Perform. Evaluation*, vol. 152, p. 102238, 2021.

[20] W. Dai, K. Foerster, D. Fuchssteiner, and S. Schmid, "Load-optimization in reconfigurable networks: Algorithms and complexity of flow routing," *SIGMETRICS Perform. Evaluation Rev.*, vol. 48, no. 3, pp. 39–44, 2020.

[21] J. Zheng *et al.*, "Dynamic load balancing in hybrid switching data center networks with converters," in *ICPP*. ACM, 2019, pp. 11:1–11:10.

[22] V. V. Vazirani, *Approximation algorithms*. Springer, 2001.

[23] H. Liu *et al.*, "Scheduling techniques for hybrid circuit/packet networks," in *CoNEXT*. ACM, 2015, pp. 41:1–41:13.

[24] S. B. Venkatakrishnan *et al.*, "Costly circuits, submodular schedules and approximate carathéodory theorems," in *SIGMETRICS*. ACM, 2016, pp. 75–88.

[25] Calient, "Edge 640 optical circuit switch," https://www.calient.net/products/edge640-optical-circuit-switch/, 2018.

[26] Polatis, "Series 6000n network optical matrix switch," https://www.hubersuhner.com/en/documents-repository/technologies/pdf/data-sheets-optical-switches/polatis-series-6000n, 2019.

[27] N. H. Azimi *et al.*, "Firefly: a reconfigurable wireless data center fabric using free-space optics," in *SIGCOMM*, 2014.

[28] D. Halperin *et al.*, "Augmenting data center networks with multi-gigabit wireless links," in *SIGCOMM*. ACM, 2011, pp. 38–49.

[29] K. Han *et al.*, "RUSH: routing and scheduling for hybrid data center networks," in *INFOCOM*. IEEE, 2015, pp. 415–423.

[30] Z. Yang *et al.*, "Achieving efficient routing in reconfigurable dcns," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 3, no. 3, Dec. 2019.

[31] D. P. Williamson and D. B. Shmoys, *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

[32] P. M. Białoń, "A randomized rounding approach to a k-splittable multicommodity flow problem with lower path flow bounds affording solution quality guarantees," *Telecom. Syst.*, vol. 64(3), p. 525–542, 2017.

[33] P. Alimonti and V. Kann, "Hardness of approximating problems on cubic graphs," in *Algorithms and Complexity*, G. Bongiovanni, D. P. Bovet, and G. Di Battista, Eds. Springer Berlin Heidelberg, 1997, pp. 288–298.

[34] M. Martens and M. Skutella, "Flows on few paths: Algorithms and lower bounds," *Networks*, vol. 48, no. 2, pp. 68–76, 2006.

[35] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[36] S. Kandula *et al.*, "The nature of data center traffic: measurements & analysis," in *IMC*, 2009.

[37] A. Roy *et al.*, "Inside the social network's (datacenter) network," in *Comput. Commun. Rev.*, vol. 45, no. 4, 2015.

[38] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proc. IMC*, 2010.

[39] C. Avin *et al.*, "On the complexity of traffic traces and implications," in *Proc. ACM SIGMETRICS*, 2020.

[40] "Characterization of the doe mini-apps," portal.nersc.gov/project/CAL/doe-miniapps.htm, 2016.

[41] M. Alizadeh *et al.*, "pfabric: minimal near-optimal datacenter transport," in *SIGCOMM*, 2013.

[42] A. Singla *et al.*, "Jellyfish: Networking data centers randomly," in *NSDI*. USENIX, 2012.

[43] W. M. Mellette *et al.*, "Rotornet: A scalable, low-complexity, optical datacenter network," in *SIGCOMM*. ACM, 2017.

[44] W. M. Mellette *et al.*, "Expanding across time to deliver bandwidth efficiency and low latency," in *NSDI*, 2020.

[45] H. Ballani *et al.*, "Sirius: A flat datacenter network with nanosecond optical switching," in *SIGCOMM*. ACM, 2020, pp. 782–797.

[46] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *SIGCOMM*, 2008.

[47] A. Valadarsky, G. Shahaf, M. Dinitz, and M. Schapira, "Xpander: Towards optimal-performance datacenters," in *CoNEXT*, 2016.

[48] S. Schmid, C. Avin, C. Scheideler, M. Borokhovich, B. Haeupler, and Z. Lotker, "Splaynet: Towards locally self-adjusting networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1421–1433, 2016.

[49] L. Poutievski *et al.*, "Jupiter evolving: transforming google's datacenter network via optical circuit switches and software-defined networking," in *ACM SIGCOMM*, 2022, pp. 66–85.

[50] G. Porter *et al.*, "Integrating microsecond circuit switching into the data center," in *SIGCOMM*, 2013.

[51] H. Ballani *et al.*, "Sirius: A flat datacenter network with nanosecond optical switching," in *SIGCOMM*. ACM, 2020, pp. 782–797.

[52] M. Dinitz and B. Moseley, "Scheduling for weighted flow and completion times in reconfigurable networks," in *INFOCOM*. IEEE, 2020.

[53] K.-T. Foerster, T. Marette, S. Neumann, C. Plant, Y. Sadikaj, S. Schmid, and Y. Velaj, "Analyzing the communication clusters in datacenters," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 3022–3032.

[54] S. Jia *et al.*, "Competitive analysis for online scheduling in software-defined optical WAN," in *INFOCOM*. IEEE, 2017, pp. 1–9.

[55] M. Nance-Hall *et al.*, "Improving scalability in traffic engineering via optical topology programming," *IEEE Trans. Netw. Serv. Manag.*, 2024.

[56] K. Foerster *et al.*, "Optflow: A flow-based abstraction for programmable topologies," in *SOSR*. ACM, 2020, pp. 96–102.

[57] N. R. Schiff *et al.*, "Chopin: Combining distributed and centralized schedulers for self-adjusting datacenter networks," in *OPODIS*, 2022.

[58] L. Luo *et al.*, "Optimizing multicast flows in high-bandwidth reconfigurable datacenter networks," *J. Netw. Comput. Appl.*, vol. 203, 2022.

[59] T. Fenz *et al.*, "On efficient oblivious wavelength assignments for programmable wide-area topologies," in *ANCS*. ACM, 2021, pp. 38–51.

[60] J. Augustine *et al.*, "Shortest paths in a hybrid network model," in *SODA*. SIAM, 2020, pp. 1280–1299.

[61] J. Castenow, C. Kolb, and C. Scheideler, "A bounding box overlay for competitive routing in hybrid communication networks," in *ICDCN*. ACM, 2020, pp. 14:1–14:10.

[62] M. Feldmann, K. Hinnenthal, and C. Scheideler, "Fast hybrid network algorithms for shortest paths in sparse graphs," in *OPODIS*, 2020.

[63] S. Coy et al., "Near-shortest path routing in hybrid communication networks," in *OPODIS*, 2021, pp. 11:1–11:23.

[64] F. Kuhn and P. Schneider, "Computing shortest paths and diameter in the hybrid network model," in *PODC*. ACM, 2020, pp. 109–118.